

Faculdade de Engenharia da Universidade do Porto



Algoritmos heurísticos para a afetação de veículos a reservas em empresas de “*rent-a-car*”

Ana Graça Morais de Oliveira Ribeiro

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Prof. José Fernando Oliveira

Co-Orientador: Eng.^a Beatriz Brito Oliveira

21 de Julho de 2016

Resumo

Em empresas de *rent-a-car*, a maximização da utilização da frota é um dos principais desafios a superar. Com um mercado competitivo e com clientes cada vez mais exigentes, surge a necessidade de disponibilizar uma frota heterogénea, nas quantidades ideais em cada estação de aluguer.

O objetivo deste projeto passa por planificar o calendário de utilização dos veículos (isto é, alocar as reservas a veículos específicos) de forma a maximizar o lucro, não comprometendo os detalhes do pedido para garantir a satisfação do cliente.

Devido à flexibilidade do cliente de escolher o início e o fim da reserva em qualquer estação à escolha, a frota de veículos deve ser partilhada entre as estações e o seu planeamento deverá ser feito de forma integrada, de forma a garantir disponibilidade no momento e local pretendido pelo cliente. A partilha e constante movimentação da frota entre as diferentes estações obriga então considerar a transferência dos veículos em vazio, que poderá ser feita utilizando um camião para transportar vários veículos de uma só vez ou através de um motorista responsável por colocar o veículo da estação pretendida. O custo destas transferências pode comprometer o lucro total alcançado e, conseqüentemente, o sucesso da empresa.

Através da aplicação e adaptação do algoritmo genético BRKGA (*Biased-Random Key Genetic Algorithm*) a este problema, procura-se neste trabalho melhorar o processo de alocação veículo-reserva, contemplando todas as restrições inerentes a este tipo de processo de alocação. Serão ainda testadas diferentes instâncias de forma a obter resultados comparáveis com os já presentes na literatura, validando e avaliando a *performance* do método desenvolvido.

Comparando os resultados obtidos utilizando o *BRKGA* com os resultados presentes na literatura verificou-se uma considerável redução do tempo de processamento em cerca de 25 vezes, encontrando-se os valores da função objetivo muito próximos dos obtidos pelos autores, embora um pouco inferiores.

Abstract

In car rental, maximizing the utilization of the fleet is a major challenge to overcome. The competitive market and increasingly demanding customers bring the need to provide a heterogeneous fleet with the right quantity of vehicles in each rental station.

The main goal of this project is to build a schedule for the vehicles' utilization (i.e., to assign reservations to specific vehicles) with the goal of maximizing profit, while complying with the order details to ensure customer satisfaction.

Due to customer flexibility in choosing the start and end of reservation at any rental station, the vehicle fleet and its planning should be shared between stations. This should be done in an integrated manner, so as to guarantee availability at the time and location desired by the customer. The fleet sharing and its constant movement between stations brings the need to consider vehicle empty transfers, which can be done by truck or by driver. The cost of those empty transfers can compromise the total profit achieved and hence the success of the company.

Applying and adapting the genetic algorithm BRKGA (Biased Random-Key Genetic Algorithm) to this problem, this work aims to improve the vehicle-reservation assignment, considering all the requirements and constraints inherent to this type of process allocation. Different instances will be tested in order to obtain comparable results with the ones already published in the literature, validating and evaluating the performance of the developed method.

This approach was compared with the results already presented in literature, and it allows the reduction of the processing time by approximately 25 times, while maintaining the value of the profit almost unchanged.

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus orientadores, Prof. Doutor José Fernando Oliveira e Prof.^a Beatriz Oliveira. Obrigada por me darem a oportunidade de fazer parte deste projeto, por tudo aquilo que me ensinaram com tanta paciência, empenho e dedicação. Além de professores, tornaram-se, para mim, modelos a seguir.

Uma palavra, em particular, para a Prof.^a Beatriz. Para além de ter sido a melhor orientadora a que alguém poderia aspirar, revelou-se uma pessoa excecional em todos os aspetos. A nível académico, pela sua inteligência, perspicácia, elevados conhecimentos e absoluta disponibilidade. A nível pessoal, pela sua amabilidade, valores e forma de ser e estar.

À Prof.^a Maria Antónia Carravilla, obrigada por me aceitar na equipa e por todo o conhecimento partilhado ao longo deste projeto.

Ao CEGI, por me acolherem com tanta simpatia e amizade e por toda a ajuda sempre prontamente disponibilizada. É realmente motivante trabalhar com uma equipa tão inteligente e fantástica. Um especial obrigada à Sara, à Maria e à Elsa, pela amizade e apoio desde o primeiro dia.

À Dra. Rosário e ao Dr. Isidro estou grata por todas as oportunidades que me foram dadas e ao Pedro Galvão pela eterna paciência no tanto que me ensinou. Estou, sobretudo, agradecida pela amizade. Ainda uma referência ao Nuno Sousa, que, há muito tempo, me apontou o rumo a seguir.

Gostaria ainda de agradecer à minha madrinha do Porto. Nunca pensei conhecer uma pessoa tão maravilhosa e generosa, com um coração tão bom e tão grande. Obrigada pela proteção, por todos os conselhos, por ser um exemplo para mim. Agora que nos conhecemos, não a largarei mais.

Finalmente, mas não menos importante, à minha família. Mãe, Pai, obrigada por todo o apoio durante estes anos, por acreditarem em mim e me darem tantas oportunidades na vida. Dou graças pelo meu irmão Ricardo, por ser a minha força, o meu maior orgulho - serei eternamente grata por ser sua irmã. À maninha do coração, Patrícia Gonçalves, que também já é parte desta família, obrigada por acreditar sempre em mim e por me dar tanta força.

Ana Graça Ribeiro

*Depois de escalar uma montanha muito alta,
descobrimos que há muitas outras por escalar.*

Nelson Mandela

Conteúdo

1	Introdução	1
1.1	Funcionamento do negócio de <i>rent-a-car</i>	1
1.2	Estruturação dos problemas de gestão de frota	2
1.3	Processo de atribuição de reserva a veículos	3
2	Revisão de Literatura	5
2.1	Problemas de gestão de frota em <i>rent-a-car</i>	5
2.2	Casos de estudo com abordagens semelhantes	8
2.3	BRKGA	9
3	Desenvolvimento do método de resolução	13
3.1	Descrição detalhada do problema	13
3.1.1	Modelo Matemático	14
3.2	Desenvolvimento e adaptação do <i>framework</i> do BRKGA	17
3.2.1	<i>Fitness</i> e Admissibilidade da solução	18
3.2.2	Descodificação dos Cromossomas	20
3.3	Resolução em horizonte rolante	23
3.3.1	Resultados preliminares – Instância Completa	23
4	Testes e Resultados	25
4.1	Análise de sensibilidade	25
4.2	Resultados Computacionais	28
5	Conclusão	37
A	Anexo: Indicadores secundários de análise	39

Lista de Figuras

2.1	BRKGA	11
3.1	Fluxograma do BRKGA	17
3.2	Codificação – Exemplo atribuição chave aleatória	21
3.3	Compartimentação do intervalo	21
3.4	Atribuição veículo-reserva	21
3.5	Codificação – Exemplo atribuição chave aleatória	23
3.6	Codificação – Exemplo de ordenação	23

Lista de Tabelas

3.1	Resultados preliminares para resolução de uma instância completa	24
4.1	Função Objetivo para $p = 500$	26
4.2	Função Objetivo para $p = 1000$	26
4.3	Função Objetivo para $p = 2500$	26
4.4	Tempo de processamento (min) para $p = 500$	27
4.5	Tempo de processamento (min) para $p = 1000$	27
4.6	Tempo de processamento (min) para $p = 2500$	27
4.7	Número de reservas alocadas para $p = 500$	28
4.8	Número de reservas alocadas para $p = 1000$	28
4.9	Número de reservas alocadas para $p = 2500$	29
4.10	Características das instâncias	30
4.11	Resultados para número máximo de gerações MaxGens=50	30
4.12	Resultados para o número máximo de gerações MaxGens=1000	31
4.13	Taxa de variação de MaxGens=50 para MaxGens=1000	32
4.14	Geração média na qual foi alcançada a melhor solução	33
4.15	Comparação do valor da Função Objetivo com Oliveira et al. (2014)	34
4.16	Comparação do tempo de processamento com Oliveira et al. (2014)	35
A.1	Tempo de transferência em vazio para $p = 500$	39
A.2	Tempo de transferência em vazio para $p = 1000$	39
A.3	Tempo de transferência em vazio para $p = 2500$	39
A.4	Comparação do número de reservas alocadas com Oliveira et al. (2014)	40
A.5	Comparação do tempo de transferência em vazio com Oliveira et al. (2014)	41

Abreviaturas e Símbolos

API	<i>Application Programming Interface</i>
BRKGA	<i>Biased Random Key Genetic Algorithm</i>
MIP	<i>Mixed Integer Programming</i>
TSP	<i>Travelling Salesman Problem</i>

Capítulo 1

Introdução

O negócio do aluguer de viaturas tem vindo ao longo dos anos a sofrer um crescimento considerável e a conquistar uma posição cada vez mais de relevo no mercado ([Pachon et al., 2006](#)). Este crescimento tem sido impulsionado pelo desenvolvimento e crescimento do sector automóvel, que atualmente permite a aquisição e manutenção de uma frota heterogénea a baixos custos, com flexibilidade e mobilidade da frota oferecida. Também a crescente adesão da população ao meio de transporte aéreo para viagens de lazer (com voos *low-cost* e larga oferta de destinos e horários) impulsionou o aluguer de veículos neste contexto, ao invés de utilização de outros transportes públicos ou mesmo transporte em veículo-próprio. Por exemplo, no caso de uma viagem Porto-Faro, atualmente, é mais económico fazer a viagem de avião em voo *low-cost* e, quando no destino, proceder ao aluguer de um veículo.

Ora, com um público cada vez mais diversificado e exigente, num mercado altamente competitivo, surgiu a necessidade de adaptação aos requisitos impostos pelos clientes. Isto constitui um problema complexo ao nível da gestão do negócio, com aumento elevado de fatores condicionantes do sucesso.

O processo de atribuição veículo-reserva em empresas *rent-a-car* é pouco abordado na literatura, pelo que, para uma melhor compreensão do problema, se estudaram casos semelhantes noutros setores, como transporte e reposição de vagões ferroviários.

Pretende-se neste estudo implementar o método BRKGA (*Biased Random-Key Genetic Algorithm*) para o processo de alocação reserva-automóvel numa empresa de *rent-a-car*. Esta meta-heurística foi já aplicada em problemas de alocação noutros contextos, tendo obtido resultados promissores ao nível da qualidade das soluções obtidas e dos tempos de processamento.

1.1 Funcionamento do negócio de *rent-a-car*

De acordo com ([Oliveira et al., 2014](#)), o cliente terá oportunidade de efetuar uma reserva de um veículo, com características específicas, num ponto de recolha/entrega de

acordo com a preferência de cada um. Tal facto obriga então a empresa a enveredar por uma organização de funcionamento coletivo de estações de aluguer em questão. Como descrito em (Pachon et al., 2003), uma das formas de estruturação do funcionamento coletivo das diversas estações consiste em introduzir neste negócio o conceito de *pool*, ou seja, agrupamento de estações de acordo com a sua situação geográfica e perfil de procura, de forma a partilharem a mesma frota de veículos.

Cada *pool* é constituída por uma determinada frota, estabelecida segundo as necessidades das estações que a constituem. Essa frota é constituída por dois tipos de veículos:

- *Veículos de gama média/baixa* – disponíveis em todas as estações em quantidades suficientes para satisfazer imediatamente a procura das reservas.
- *Veículos especiais* – gamas com características particulares (como monovolumes, todo-o-terreno, alta cilindrada, blindados, entre outras que devido às suas características especiais apenas se encontram disponíveis em determinadas estações de aluguer, em quantidades reduzidas. São, portanto, transportados frequentemente entre estações, em vazio, para assegurar a satisfação de reservas dentro da *pool*.

1.2 Estruturação dos problemas de gestão de frota

Segundo (Pachon et al., 2006), o problema de gestão de frota em empresas rent-a-car está dividido em três fases: *i) pool segmentation*, *ii) strategic fleet planning* e *iii) tactical fleet planning*.

- i *pool segmentation* – de acordo com a localização geográfica e com o histórico de procura em cada uma das estações, estas são agregadas por forma a ser possível a partilha da frota. Frequentemente os gestores da zona contribuem com a sua experiência no processo de agregação.
- ii *strategic fleet planning* – a decisão a tomar é a quantidade de veículos a atribuir a cada *pool*. Enfrenta como condicionante o facto de os gestores de frota centrais não estarem muitas vezes em consenso com os gestores de frota locais.
- iii *tactical fleet planning* – nesta fase é estabelecido o número de veículos a atribuir a cada estação, dentro de cada *pool* num determinado período. É especialmente importante devido ao facto de incluir o transporte de veículos, em vazio, que pode ser efetuado em conjunto com recurso a camiões de transporte, ou individual por um motorista atribuído exclusivamente para a tarefa.

Acrescenta-se ainda o seguinte problema:

- iv *vehicle-reservation assignment* – nesta fase pretende-se estudar a alocação automóvel-reserva.

1.3 Processo de atribuição de reserva a veículos

Sendo a procura imprevisível e sujeita a diversos fatores externos, como épocas festivas ou eventos sociais relevantes, é importante conhecer com a maior antecedência possível o fluxo de reservas. Para tal, os pedidos antecipados de reservas são normalmente tratados de forma particular, sendo oferecidas condições especiais como descontos no preço do serviço.

De forma a combinar e tratar devidamente os dados das reservas, estas devem conter parâmetros específicos como: data e estação de levantamento do veículo, data e estação de entrega do veículo, receita obtida, tipo de veículo pretendido pelo cliente e prioridade do serviço (as reservas confirmadas pelo cliente têm prioridade em relação às não confirmadas). De facto, cada reserva deve ainda ser confirmada pelo cliente sob pena de não ser efetuada, pois perde a prioridade de alocação face às restantes.

Relativamente aos veículos, devem ser caracterizados por um número de identificação e pelo seu estado de ocupação. Os tempos de inatividade devem ainda ser contemplados, e neste estado, podem ser considerados como inatividade provisória, que se denomina *impros*, ou inatividade permanente. Inatividade provisória, tal como o próprio nome indica, tem uma curta duração e pode ocorrer devido a limpeza do veículo, reparação mecânica, entre outros. Inatividade permanente verifica-se quando o veículo é excluído definitivamente da frota, não estando mais disponível para associação a reservas. Sendo a frota heterogénea, existem veículos cujas características especiais (4x4, caixa-aberta, comerciais, veículos de luxo) apenas permitem a existência de algumas unidades, tornando-se particularmente importante a calendarização otimizada das suas reservas.

Durante o planeamento, ou no ato de *check-out*, mesmo mediante um planeamento cuidado das alocações, pode surgir a impossibilidade de atribuição de um carro do tipo escolhido. Nessas situações, ocorrerá a oportunidade de ser feito, mediante a aprovação do cliente, o *upgrade* ou em último caso o *downgrade* do veículo. Na situação de *upgrade*, será atribuído ao cliente um veículo de gama superior ou com características especiais, cobrando o mesmo valor. Já na situação de *downgrade*, o cliente será confrontado com a possibilidade do seu pedido ser satisfeito com um veículo de gama inferior, tendo no entanto uma redução do preço. De salientar que nenhuma alteração à gama pretendida proposta pelo sistema de planeamento ou elemento do *staff*, é efetuada sem prévia aceitação do cliente.

Capítulo 2

Revisão de Literatura

O processo de aluguer de veículos exige a combinação de vários processos. Sendo este um setor próspero com acentuado crescimento, uma melhoria na eficácia permitirá um aumento significativo da rentabilidade (Pachon et al., 2006). Este documento centrar-se-á, fundamentalmente, no processo de atribuição de reservas a veículos especiais. Modelos de programação inteira mista que permitem a obtenção de boas soluções do problema de alocação de um veículo a uma reserva já foram desenvolvidos (Oliveira, 2013), sendo, no entanto, ainda vantajosa a implementação de métodos diferentes, de forma a testar alternativas e possíveis melhorias, não apenas na solução obtida, mas também ao nível do tempo de processamento ou da resolução de problemas de maiores dimensões.

Para melhor perceção do problema, consideraram-se estudos com diferentes abordagens e contextos. Também o ano de publicação se tornou relevante, pois avanços têm sido feitos ao longo do tempo.

Inicialmente, foram considerados artigos relacionados com o processo de aluguer de veículos, especialmente automóveis, e distribuição da respetiva frota. Como é ainda um tema pouco aprofundado (Oliveira, 2013), complementou-se a revisão com problemas de atribuição semelhantes, como os dos vagões ferroviários cujo principal problema é também a reposição de frota em vazio. No final de cada secção, será feita uma discussão por forma a perceber a contribuição no caso em estudo.

2.1 Problemas de gestão de frota em *rent-a-car*

O processo de atribuição veículo-reserva apenas é possível se o veículo solicitado pelo cliente estiver disponível na estação pretendida. Pachon et al. (2006) estudam especialmente a distribuição da frota de veículos através das diferentes estações. Para o seu estudo, estruturam o problema em 3 partes: a *pool segmentation*, onde ocorre o agrupamento das estações de aluguer que irão compartilhar frota, é feita de acordo com a localização geográfica das estações e nos dados de anos anteriores de procura; *strategic fleet planning*, que consiste na atribuição de uma determinada quantidade de veículos a cada *pool*; *tactical*

fleet planning, onde se detalha concretamente quantos veículos de cada tipo serão atribuídos a cada uma das estações da *pool* correspondente. Esta última fase é abordada detalhadamente em Pachon et al. (2003) onde é dada especial ênfase ao transporte de veículos em vazio. Devido à complexidade do problema, os autores decompõem em dois sub-problemas, i) desdobramento da frota, e ii) transporte em vazio. Segundo os autores, o processo de desdobramento da frota é feito verificando a disponibilidade de veículos no final de cada dia, permitindo então a posterior decisão de transporte de veículos em vazio; transporte esse feito durante a noite para posterior disponibilidade no dia seguinte no local adequado. Este problema foi resolvido recorrendo à implementação de uma heurística, que permitiu obter bons resultados de custos de transporte de veículos, considerando condições próximas à realidade.

Após a alocação adequada da frota às estações, surge a necessidade de atribuição veículo-reserva. Oliveira et al. (2014) utilizam uma heurística *relax-and-fix* para o processo de alocação de reservas. Aqui, surge uma nova restrição - a frota considerada possui dois tipos de veículos: os *free sale vehicles* e os veículos especiais. Os *free sale vehicles* são veículos *standard* sem características especiais, existindo em grande quantidade em cada estação de aluguer, sendo portanto a sua alocação a uma reserva possível de efetuar imediatamente. Já para os veículos especiais, com características particulares (dois lugares, blindados, alta cilindrada, carrinhas de caixa-aberta, entre outros) e que, devido a essas particularidades não existem em larga escala, a sua alocação à reserva pretendida é dificultada, tornando o processo de transporte em vazio crucial. O estado de ocupação dos veículos permite sinalizar inatividade provisória (*impros*) ou permanente dos veículos. A possibilidade de *check-in* e *check-out* em estações diferentes dentro da mesma *pool* são permitidas. Os autores consideram ainda que os pedidos do cliente devem ser atendidos, se possível, com o grupo exigido inicialmente mas, se não for possível, a opção de *upgrade* e, em último caso, *downgrade*, deve ser colocada. Em caso de *upgrade*, será atribuído um veículo de gama superior para ocupar a reserva pelo mesmo preço. Em caso de *downgrade*, um veículo de gama inferior deve ser proposto, no entanto, com diminuição do preço. É considerado ainda que reservas que já foram confirmadas aos clientes têm de ser cumpridas.

Li e Tao (2010) centram o seu estudo na alocação da frota às estações, e apenas consideram uma *pool* composta por duas cidades, com a transferência de veículos em vazio feita durante o período da noite. A procura surge através de dois tipos de clientes: os clientes de viagem única, que fazem *check-out* e *check-in* em estações distintas e que desequilibram o balanceamento da frota, e os clientes de viagem de ida-e-volta, onde o *check-in* e o *check-out* são efetuados na mesma estação, sendo estes prioritários no processo de alocação de reservas. Os autores recorreram, então, à implementação de um modelo de programação dinâmica que se centra inicialmente no cálculo do tamanho ótimo da frota necessária em cada local, e posteriormente no transporte em vazio dos veículos, de forma a satisfazer as necessidades inicialmente calculadas. Este modelo proposto tem em conta que há reservas que não será possível satisfazer e considera que a procura nas diferentes

estações é uma variável aleatória.

Uma importante abordagem relativamente a *tactical fleet planning* é feita por [Fink e Reiners \(2006\)](#), que abordam o problema através de *minimum cost network flow optimization*. Os autores concluíram ainda que, para a melhoria do processo de alocação de veículos, o fator-chave é a existência de um sistema de informação com dados completos correspondentes ao estado de ocupação dos veículos aliado a um modelo capaz de prever a procura a curto-prazo (de alguns dias até duas semanas). Aqui, a transferência de veículos em vazio torna-se novamente relevante. Relativamente ao processo de alocação, os autores não recomendam a alocação central de veículos, uma vez que leva a uma maior complexidade nos sistemas de informação, mais suscetível a ocorrência de falhas. Também são aqui consideradas cruciais as duas fontes de informação mais importantes: os clientes e o *staff*. Como o objetivo se centra na previsão a curto-prazo da procura, os clientes devem ser incentivados a fazer reservas com alguma antecedência. Para tal, o processo de reserva deve ser feito com alguma facilidade, introduzindo como incentivo possibilidades, por exemplo, da ocorrência de *upgrades* (que, em caso de alugueres feitos na hora, não é possível, estando o cliente sujeito à não disponibilidade do veículo pretendido). O sistema mede ainda a performance da estação de acordo com os valores do lucro obtido e dos custos de manutenção dos veículos.

[Beaujon e Turnquist \(1991\)](#) centram o seu estudo na identificação de métodos e estratégias que permitam o adequado dimensionamento da frota de aluguer (não específico para *rent-a-car*) e alocação das reservas aos veículos disponíveis. Segundo os autores, o impacto da melhoria da utilização dos recursos disponíveis é, neste caso, muito mais importante do que os custos associados ao processo de transporte de veículos, não colocando em causa a qualidade do serviço. Assume-se, então, o processo de transporte em vazio como o ponto fulcral do problema, sendo obrigatório satisfazer uma procura imprevisível, o que no caso em estudo poderia condicionar o desempenho do sistema na alocação de reservas.

Através do método de relaxação *Lagrangeana*, [Ernst et al. \(2011\)](#) pretendem *re-otimizar* diariamente o calendário de alocação de reservas. Para isso, são utilizadas uma heurística e um modelo que, em conjunto, permitem a obtenção de resultados favoráveis em intervalos de tempo relativamente reduzidos. Os autores consideram o tempo importante pois, uma vez que o processo de reservas é contínuo e imprevisível, o sistema não se pode encontrar em manutenção mais do que 1 hora; daí a velocidade de *re-otimização* do calendário ser muito importante. O modelo desenvolvido inclui veículos de tipos distintos, possíveis de alugar em várias estações, sendo permitida a mudança do tipo de veículo se o desejado pelo cliente não estiver disponível (com devida penalização na margem de lucro) - *upgrade*.

Também [Yaghini e Khandaghabadi \(2013\)](#) estudam o problema de dimensionamento da frota através da combinação de meta-heurísticas como algoritmos genéticos e arrefecimento simulado (*Simulated Annealing*), alcançando resultados positivos. Os autores estruturam então o problema em duas partes: a alocação da frota disponível às estações e a determinação do tamanho ótimo da frota. Assumem ainda que a procura é conhecida.

Já [Carroll e Grimes \(1995\)](#) apresentam a sua análise para a tomada de algumas decisões, nomeadamente qual o tamanho da frota a adquirir, considerando planos de compra de novos automóveis, remoção dos antigos e determinação de cada tipo de veículo. Os autores consideram a alocação da frota às estações, transporte em vazio dentro ou fora da *pool*, tendo em conta eventos que desencadeiam o aumento da procura. Outras questões incluem o produto a oferecer (preço e descontos admitidos), qual o produto a disponibilizar e o planeamento da distribuição da frota.

Comentários: Na maior parte dos estudos, os autores focam a sua análise no processo de alocação às estações dos veículos para os locais onde surge a procura, tarefa que se revela particularmente complexa devido à necessidade de previsão de uma procura variável e aleatória. A abordagem mais utilizada é de divisão de estações por *pools*, sendo que estar dentro da mesma *pool* implica partilha da mesma frota heterogénea, com diferentes quantidades de veículos conforme o grupo a que pertencem (veículos com qualidades particulares existem normalmente em menor número do que veículos ditos comuns). No entanto, ao longo dos artigos analisados apenas [Oliveira et al. \(2014\)](#) e [Ernst et al. \(2011\)](#) aprofundam o tema de alocação de veículos a reservas detalhadamente. De facto, após a determinação do tamanho (*fleet sizing*) da frota e a determinação do número de veículos que devem ser repostos em que estações (*fleet deployment*), onde também se decide quais são as movimentações em vazio a curto-prazo necessárias, surge a necessidade de considerar o problema de alocação reserva-veículo (*vehicle-job assignment*) onde, mediante os custos de transferências em vazio e disponibilidade, se decide qual o veículo que será alocado a cada pedido. O tempo de processamento do processo de alocação reserva-veículo é aqui importante devido à necessidade de atualização frequente da calendarização das marcações.

2.2 Casos de estudo com abordagens semelhantes

Um dos problemas que aborda o processo de atribuição é o caso do transporte em vagões/contentores ferroviários. [Bojović \(2002\)](#) foca o seu estudo na determinação do tamanho da frota e na sua alocação às estações, considerando para o estudo uma frota homogénea. Os autores dividem então o estudo do problema em duas áreas: a primeira foca-se na determinação da capacidade de transporte em vazio para os diferentes destinos; prossegue-se com a determinação do tamanho ótimo da frota, fator que é determinante para o tráfego das ferrovias. O modelo apresentado, baseado em Controlo Estocástico, demonstrou-se eficaz no dimensionamento da frota e em termos de eficiência computacional, tornando possível balancear o custo da procura desconhecida com os custos totais de utilização e posse dos vagões.

[Spieckermann e Voß \(1995\)](#) estudam o caso de uma empresa em concreto, abordando o problema da distribuição dos vagões ferroviários em vazio. Após comparações de vários

métodos, a abordagem através da heurística de *First-Fit* permitiu a melhoria no atraso de transporte de vagões; no entanto, apresenta várias limitações, especialmente ao nível do tempo de processamento do algoritmo. Para a ocorrência de uma reserva, o cliente deve contactar o expedidor e informá-lo do número de vagões de que necessita, o tipo (opcional), a data de entrega e o local. Com a utilização de dados reais, considerando todos os pedidos, os autores concluíram que a utilização de um sistema de processamento central seria demasiado lento, sendo necessário correr os algoritmos localmente em cada estação, permitindo apenas a utilização de um pequeno número de vagões.

Milenković et al. (2015) utilizam um modelo de controlo preditivo para o desenvolvimento de um modelo dinâmico, que tem como objetivo a otimização da alocação de uma frota heterogénea de vagões ferroviários. Os autores consideram que a incerteza da procura é crescente ao longo do tempo, ou seja, quanto mais distante for o período a considerar, mais imprevisível será a sua procura. Crainic (2000) direciona a sua análise para o sistema de transporte de cargas em geral, não se focando no processo de alocação da reserva ao pedido, mas na otimização do processo de transporte de forma a chegar ao local em tempo útil para o cliente.

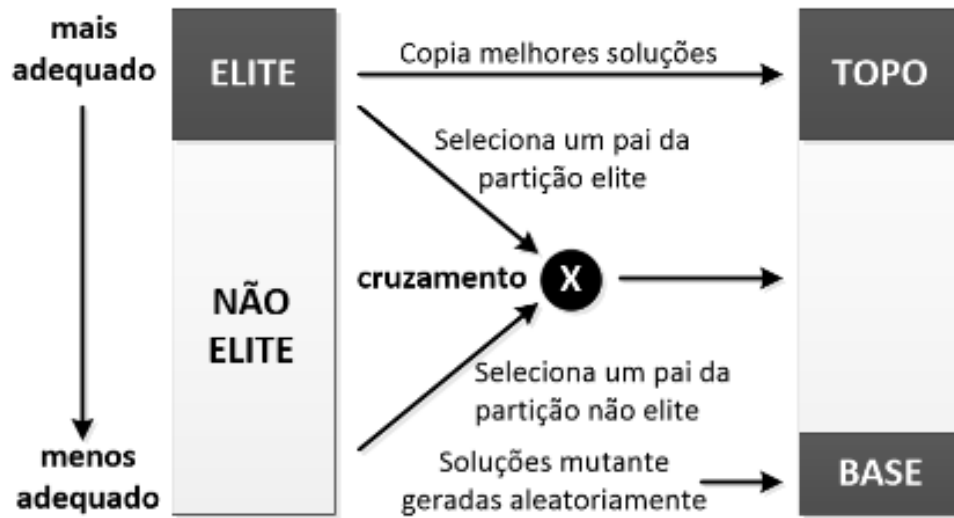
Focando-se também no processo de transporte de mercadorias, Crainic et al. (1984) estudam o problema de alocação de serviços ao transporte ferroviário de mercadorias. Apesar de incluir problemas que tratam o processo de atribuição, este caso de estudo não se adequa particularmente ao problema em estudo pois considera na implementação da sua heurística apenas o problema de transporte.

Comentários: O caso de transporte de vagões ferroviários assemelha-se ao caso de *rent-a-car* devido à necessidade de resolução do processo de atribuição veículo-reserva. Tem como objetivo a satisfação de todas as reservas, com o mínimo atraso possível. Um dos principais problemas de ambas as indústrias é o transporte em vazio. Com custos de posse e transporte superiores, é igualmente crítica a sua disponibilidade no local adequado para satisfazer a procura do cliente. No entanto, neste caso é aceitável a consideração de atrasos para a redução de custos, enquanto que no caso das empresas de *rent-a-car* o atraso pode significar a insatisfação ou mesmo perda do cliente.

2.3 BRKGA

Com vista à melhoria nos tempos de processamento e computação, visando a possibilidade de resolver problemas maiores (mais realistas) e, se possível, conseguir melhores resultados, investigou-se que contribuição o método BRKGA poderia trazer para o caso de alocação veículo-reserva em empresas *rent-a-car*. Gonçalves e Resende (2010) apresentam o método BRKGA (*Biased Random-key Genetic Algorithms*) como uma variação do RKGA (*Random-key Genetic Algorithms*) e explicam detalhadamente o seu funcionamento. Uma população é constituída por vários indivíduos (soluções do problema). Cada

indivíduo tem uma string de n genes correspondente, denominada cromossoma. O valor de cada gene é chamado alelo. Em cada iteração, através de diferentes mecanismos, uma nova geração de indivíduos é gerada (um novo grupo de soluções) de forma elitista. Segundo os autores, o principal desafio do método BRKGA é a implementação do decodificador. O decodificador tem como função “traduzir” um vetor de números reais (que representa um indivíduo) numa solução para o problema de alocação veículo-reserva, e vice-versa, permitindo, assim, avaliar a qualidade da solução (valor da função objetivo) e ainda a sua admissibilidade. Esta é a parte dependente do problema, complementando a parte independente que é unicamente o processo de pesquisa de soluções no hipercubo. O algoritmo inicia-se com a geração zero. A este grupo heterogêneo inicial, composto por p vetores de n números aleatórios entre $[0, 1]$, é aplicado o conceito de sobrevivência da melhor solução, dando origem a dois grupos de indivíduos: um pequeno grupo que possui as melhores soluções, denominado ELITE (p_e), e um grupo com os restantes valores denominado NÃO ELITE ($p_e < p - p_e$). A partir de uma geração k é gerada a geração $k + 1$ através de *crossover*, cópias e mutação. Para o processamento das operações de *crossover* é selecionado um indivíduo do grupo ELITE e um do grupo NÃO ELITE. Após o cruzamento entre os indivíduos selecionados, surgem novos cromossomas, que são avaliados mediante o processamento do decodificador, dando origem a parte da geração $k + 1$. A nova geração segue uma estratégia elitista, também copiando para o topo da ‘pilha’ as melhores soluções do grupo ELITE, seguindo-se as soluções do *crossover*, e finalizando com a introdução de cromossomas mutantes, que não são mais do que cromossomas com números aleatórios gerados, para introduzir variabilidade e ser possível explorar novas soluções (ver Figura 2.1). O procedimento é repetido até ser atingido o critério de paragem. O RKGA difere do BRKGA no processo de seleção dos progenitores. Enquanto, no primeiro, os progenitores são selecionados aleatoriamente da população total, o BRKGA faz uma seleção elitista, combinando um progenitor selecionado do grupo ELITE e outro do grupo NÃO-ELITE. Após vários testes, os autores verificaram que o método se revela bastante rápido no processo de procura da solução.

Figura 2.1: BRKGA - adaptado de [Gonçalves e Resende \(2004\)](#)

Para perceber como estruturar o decodificador para este problema, foram estudados problemas de atribuição que usavam o BRKGA como método de resolução. O estudo realizado por [Marques et al. \(2014\)](#) centra a sua análise na implementação do método BRKGA na marcação da data/sala de intervenção cirúrgica num hospital português. Aqui o objetivo é a maximização da ocupação das salas e do número de cirurgias marcadas, sendo que existem dois tipos de cirurgias: cirurgias convencionais e ambulatoriais, com prioridades diferentes. O planeamento é feito para uma semana, com cirurgias a ocorrer das 8h30 até às 20h, em salas diferentes conforme a especialidade, num total de 5 especialidades distintas. São então organizados os dados relativos a cada intervenção (data, sala hora de início) numa matriz dias \times salas, compostas por números inteiros, onde cada número inteiro corresponde a uma especialidade cirúrgica. Tendo em conta a aproximação do modelo à realidade, os autores concluíram que o método BRKGA traz vantagens não só relativamente ao tempo computacional como à qualidade das soluções obtidas, tendo sido possível incluir o a lista de espera completa. Aliado a isto, também a sua implementação permite a redução do número de funcionários, pois o projeto automatizado requer menos trabalho que o processo de alocação manual.

[Lalla-Ruiz et al. \(2014\)](#) tratam o problema de alocação de gruas de descargas nos cais de atracagem dos navios, e qual o cais onde cada navio pode atracar. Os autores consideram que cada navio tem um número variável de contentores para serem descarregados, podendo-lhes ser alocado um número variável de gruas de descarga. Quanto maior for o número de contentores a descarregar, maior número de gruas necessitará para minimizar o tempo de permanência no porto. É então utilizado o método BRKGA para tratar este problema, sendo que a nossa análise se foca uma vez mais no decodificador. Os cromossomas são divididos em duas subsequências de genes. A primeira é um vetor v constituído por uma lista ordenada de n genes (em que n é igual ao número de navios) e é utilizado para

determinar a prioridade de atracagem dos navios. A identificação de cada navio é feita de acordo com a sua posição no cromossoma não ordenado, ou seja, a primeira posição corresponde ao navio 1, a segunda posição ao navio 2, etc. A segunda parte do cromossoma tem também n genes (igual ao número de navios) e é utilizada para o processo de atribuição de perfis aos navios. Os perfis são sequências pré-determinadas de guias que operam na descarga dos navios, com um valor associado. Para a descodificação desta sequência de cromossomas divide-se então o intervalo $[0,1]$ no número de perfis P_i de cada navio, resultando em P_i sub-intervalos correspondentes para cada perfil. Se o número do alelo é encontrado em cada um desses intervalos, então o perfil correspondente é atribuído. A posição dos alelos na subsequência corresponde ao número do navio, novamente. Sendo um problema complexo e de difícil resolução recorrendo, por exemplo, ao *CPLEX*, o BRKGA permitiu a obtenção de boas soluções com tempos de computação reduzidos.

Para a implementação do BRKGA, é disponibilizada por [Gonçalves e Resende \(2014\)](#) uma API com a parte independente do problema implementada, onde apenas será necessária a implementação da parte dependente do problema, ou seja, do descodificador. Contudo, devem ser ajustados alguns parâmetros para otimização e adaptação do método ao problema em estudo. Os autores apresentam uma gama recomendada de variação dos parâmetros relativa ao tamanho da população (p), tamanho da partição da população denominada ELITE p_e , tamanho da partição da população mutante (p_m), probabilidade de herança de um gene do progenitor ELITE (ρ_a), número de iterações sem melhorias ao fim das quais deve reiniciar (k_r), número de populações paralelas (π), frequência de intercâmbio de população (k_p) e número de intercâmbio de soluções (k_m). Os autores recomendam ainda alguns critérios de paragem, como o tempo de execução do código, número máximo de iterações, número máximo de reinícios ou ainda parar mal seja encontrada a solução pretendida. Estes parâmetros devem ser adaptados de acordo com o problema em estudo.

Comentários: Mediante os casos de estudo revistos, semelhantes ao problema das empresas de *rent-a-car* aqui considerado, o método BRKGA tem-se revelado bastante eficiente a encontrar boas soluções em tempos de computação bastante reduzidos, mesmo para problemas complexos. Apesar dos outros métodos já implementados serem capazes de alcançar boas soluções para os problemas, exigem tempos de processamento demasiado elevados. No caso de estudo das empresas de *rent-a-car*, tempos de computação reduzidos são extremamente importantes, para possibilitar a alocação quase contínua de novas reservas, com o sistema atualizado.

Validação do método – implementação TSP Para melhor compreender o comportamento do BRKGA, este foi implementado pela autora deste trabalho para o problema genérico de *Travelling Salesman*. Foi assim confirmada a rapidez e eficiência na resolução do problema, utilizando instâncias padrão da literatura.

Capítulo 3

Desenvolvimento do método de resolução

3.1 Descrição detalhada do problema

O problema da atribuição de um veículo a uma reserva na empresa de *rent-a-car* em estudo é crucial para o alcance do objetivo principal – aumentar o lucro. Uma das melhorias que podem ser feitas é ao nível do calendário de marcação das reservas, procurando maximizar o estado de ocupação dos veículos. Com uma frota heterogénea e múltiplas estações de aluguer, é necessário distribuir de forma eficiente os veículos pelas estações, especialmente aqueles que existem em pequena quantidade devido às suas características mais particulares (como, por exemplo, automóveis de luxo, tração 4x4, ...). Além da escolha do tipo de veículo, o cliente tem ainda a possibilidade de iniciar a sua reserva numa determinada estação e terminar em outra distinta.

Cada veículo é caracterizado por uma identificação, que é única para cada um, a estação onde estará disponível inicialmente, a data desta disponibilidade inicial, o grupo a que pertence e a *block date*, que é a data a partir da qual ficará permanentemente indisponível. Devido à elevada procura, o veículo pretendido pelo cliente pode não se encontrar disponível, sendo então proposto um *upgrade* ou, como último recurso, um *downgrade*. O *upgrade* consiste na atribuição à reserva de um veículo considerado de gama superior, sem acréscimo ao preço da reserva. A sua atribuição é feita sem prévia aceitação do cliente pois, como se trata de um veículo de gama superior pelo mesmo valor, é razoavelmente assumido que é sempre aceite. Já no caso do *downgrade*, é proposto um veículo de gama inferior, reduzindo significativamente o valor a pagar pela reserva, tendo de existir aceitação prévia por parte do cliente.

Cada reserva é caracterizada pela data e estação de início, data e estação de fim, lucro obtido se satisfeita, estado de aceitação do cliente de *downgrade*, e o grupo do veículo pretendido. Possui, ainda, um parâmetro relativo à sua confirmação ao cliente, pois

reservas já confirmadas têm obrigatoriamente de ser cumpridas, tendo assim prioridade de alocação em relação às restantes reservas não confirmadas.

Neste tipo de negócio é ainda necessário fazer a manutenção da frota, o que causa indisponibilidade temporária do veículo, que pode ser por exemplo para limpeza, resolução de problemas mecânicos, entre outros. Este período de tempo é denominado por *impro* (Oliveira et al., 2014).

3.1.1 Modelo Matemático

O modelo seguido para a estruturação, resolução e avaliação do problema é o mesmo apresentado por Oliveira et al. (2014).

Devem então ser consideradas os seguintes parâmetros que caracterizam o conjunto de reservas \mathcal{R} , com $r \in \{1, \dots, R\}$:

- so_r estação de saída da reserva r (estação inicial);
- do_r data de saída da reservas r (instante inicial);
- si_r estação de entrada da reserva r (estação final);
- di_r data de entrada da reserva r (instante final);
- pft_r lucro da reserva r ;
- gr_r grupo pretendido na reserva r ;

$$st_r = \begin{cases} 1 & \text{se a reserva } r \text{ foi confirmada ao cliente,} \\ 0 & \text{caso contrário;} \end{cases}$$

$$dga_r = \begin{cases} 1 & \text{se a reserva } r \text{ aceita } downgrades, \\ 0 & \text{caso contrário.} \end{cases}$$

O conjunto de veículos \mathcal{V} presentes na frota, com $v \in \{1, \dots, V\}$, caracterizam-se por:

- csi_v estação de entrada do veículo v (estação de disponibilidade inicial do veículo);
- cdi_v data de entrada do veículo v (instante de disponibilidade inicial do veículo);
- gv_v grupo do veículo v ;
- bd_v *block date* do veículo v .

Os períodos de indisponibilidade temporária dos veículos, denominados por *impros*, são tratados como reserva especiais, onde a sua atribuição ao veículo em questão é obrigatória e o lucro é nulo.

$$imp_r^v = \begin{cases} 1 & \text{se a reserva } r \text{ é um } impro \text{ do veículo } v, \\ 0 & \text{caso contrário.} \end{cases}$$

Como a empresa dispõe de várias estações de aluguer e oferece a possibilidade do cliente iniciar a reserva numa estação e finalizar em outra estação distinta, devem ainda ser considerados os tempos e custos de transferências em vazio entre a estação onde se encontra o veículo e a estação onde é requisitado, s_1, s_2 :

cet_{s_1, s_2} custo da transferência em vazio entre as estações s_1 e s_2 ;

tet_{s_1, s_2} tempo de transferência em vazio entre as estações s_1 e s_2 .

Relativamente à possibilidade de satisfação da reserva com grupos distintos dos pretendidos pelo cliente, ou seja, efetuar um *upgrade* ou *downgrade* da reserva, são considerados os seguintes parâmetros:

$$\begin{aligned} s_{gr_r, gv_v} &= \begin{cases} 1 & \text{se o grupo pretendido } gr_r \text{ é igual ao grupo do veículo } gv_v \text{ } (gr_r = gv_v), \\ 0 & \text{caso contrário;} \end{cases} \\ u_{gr_r, gv_v} &= \begin{cases} 1 & \text{se o grupo pretendido } gr_r \text{ permite a possibilidade de } upgrade \text{ } gv_v, \\ 0 & \text{caso contrário;} \end{cases} \\ d_{gr_r, gv_v} &= \begin{cases} 1 & \text{se o grupo pretendido } gr_r \text{ permite a possibilidade de } downgrade \text{ } gv_v, \\ 0 & \text{caso contrário.} \end{cases} \end{aligned}$$

As variáveis de decisão definem-se por:

$$x_{ir}^v = \begin{cases} 1 & \text{se o veículo } v \text{ satisfaz as reserva } r \text{ após a reserva } i, \\ 0 & \text{caso contrário;} \end{cases}$$

Oliveira et al. (2014) propõe então o seguinte modelo MIP:

$$\begin{aligned} \max \sum_{v \in \mathcal{V}} & \left(\sum_{i=1}^R \sum_{r=1}^{R+1} (pft_r - cet_{si_i, so_r}) x_{ir}^v + \sum_{r=1}^{R+1} (pft_r - cet_{csi_v, so_r}) x_{0r}^v \right) s_{gr_r, gv_v} + \\ & \sum_{v \in \mathcal{V}} \left(\sum_{i=1}^R \sum_{r=1}^{R+1} (pft_r - cet_{si_i, so_r} - 1) x_{ir}^v + \sum_{r=1}^{R+1} (pft_r - cet_{csi_v, so_r} - 1) x_{0r}^v \right) u_{gr_r, gv_v} + \quad (3.1) \\ & \sum_{v \in \mathcal{V}} \sum_{i=0}^R \sum_{r=1}^{R+1} x_{ir}^v d_{gr_r, gv_v} \end{aligned}$$

Sujeito a:

$$\sum_{r \in \mathcal{R}} x_{0r}^v \leq 1, \forall v \in \mathcal{V} \quad (3.2)$$

$$\sum_{v \in \mathcal{V}} \sum_{i \in \mathcal{R}} x_{ir}^v \leq 1, \forall r \in \mathcal{R} \quad (3.3)$$

$$\sum_{i \in \mathcal{R}} x_{iu}^v - \sum_{r \in \mathcal{R}} x_{ur}^v = 0, \forall v \in \mathcal{V}, \forall u \in \mathcal{R} \setminus \{0, R+1\} \quad (3.4)$$

$$(-di_i - tet_{si_i, so_r} + do_r)x_{ir}^v \geq 0, \forall v \in \mathcal{V}, \forall i, r \in \mathcal{R} \quad (3.5)$$

$$x_{ir}^v \leq s_{gr_r, gv_v} + u_{gr_r, gv_v} + d_{gr_r, gv_v}, \forall v \in \mathcal{V}, \forall i, r \in \mathcal{R} \quad (3.6)$$

$$\sum_{i \in \mathcal{R}} \sum_{v \in \mathcal{V}} x_{ir}^v d_{gr_r, gv_v} \leq dga_r, \forall r \in \mathcal{R} \quad (3.7)$$

$$x_{0R+1}^v = 0, \forall v \in \mathcal{V} \quad (3.8)$$

$$(di_i - bd_v)x_{iR+1}^v \leq 0, \forall v \in \mathcal{V}, \forall i \in \mathcal{R} \quad (3.9)$$

$$\sum_{i \in \mathcal{R}} x_{ir}^v \geq imp_r^v, \forall v \in \mathcal{V}, \forall r \in \mathcal{R} \quad (3.10)$$

$$\sum_{v \in \mathcal{V}} \sum_{i \in \mathcal{R}} x_{ir}^v \geq st_r, \forall r \in \mathcal{R} \quad (3.11)$$

$$x_{ir}^v \in \{0, 1\}, \forall r \in \mathcal{R} \quad (3.12)$$

Função Objetivo

Sendo o lucro variável conforme os diferentes perfis de reserva – *upgrade*, *downgrade* ou reserva normal – a função objetivo é constituída por três parcelas, cada uma correspondente a um perfil. A primeira parcela refere-se às reservas normais, onde o grupo de veículo pretendido pelo cliente é o mesmo alocado à reserva. Neste perfil, o lucro total obtido é a diferença entre o lucro da reserva e o custo de transferência entre a estação onde se encontra o veículo e a estação de início da reserva.

Para o caso de *upgrade*, o cálculo é também a diferença entre o lucro obtido pela satisfação da reserva e o custo de transferência entre estações (se necessário). No entanto, sofre a penalização de 1 unidade monetária como forma de garantir que esta opção é menos atrativa que atribuir o veículo pedido pelo cliente, e que apenas é considerada quando tal não é possível.

Em caso de *downgrade*, apenas é acrescentada 1 unidade monetária ao lucro obtido. Apesar de tal não representar o verdadeiro lucro da empresa, consegue-se assim garantir que o *downgrade* apenas é oferecido em último recurso.

Restrições

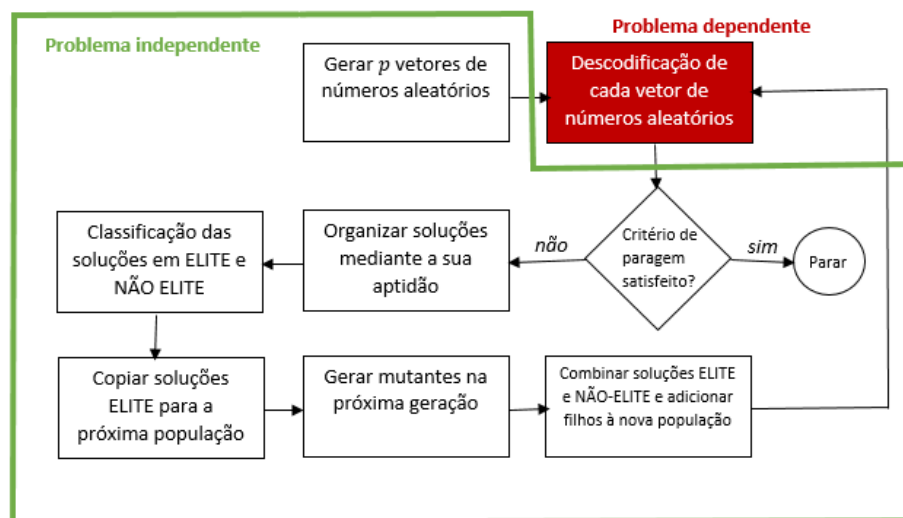
- Em 3.2 é garantido que é necessária pelo menos uma reserva para se iniciar o processo de alocação.

- Cada veículo apenas pode ter uma reserva atribuída, e cada reserva apenas um veículo (3.3).
- Para assegurar continuidade das atribuições, quando uma reserva é alocada, deve existir uma reserva anterior e uma posterior (3.4).
- Uma reserva só poderá ser alocada se a sua data de início for maior ou igual à data de finalização da anterior mais o tempo de transferência em vazio (se necessário) (3.5).
- O grupo do veículo atribuído a uma reserva corresponder ao grupo pretendido pelo cliente ou a um grupo compatível (3.6).
- Apenas pode ser feito um *downgrade* se for previamente aceite pelo cliente (3.7).
- Os nós fictícios inicial e final não pode estar diretamente ligados (3.8).
- A última reserva alocada a um veículo não pode coincidir com os períodos de indisponibilidade permanente (*block date*) (3.9).
- Nos períodos de indisponibilidade temporária dos veículos (*impros*) não pode existir nenhuma reserva alocada (3.10).
- As reservas confirmadas devem ser obrigatoriamente satisfeitas (3.11).
- Todas as variáveis são binárias (3.12).

3.2 Desenvolvimento e adaptação do framework do BRKGA

Estruturalmente, o BRKGA divide-se em duas grandes partes: uma parte independente do problema e uma parte dependente do problema.

Figura 3.1: Fluxograma do BRKGA baseado em (Grasas et al., 2014)



A parte independente encontra-se implementada numa API disponibilizada por [Silva et al. \(2015\)](#) sendo, no entanto, necessário adaptar ao problema em estudo, nomeadamente o cálculo do *fitness* e o ajuste dos parâmetros já incluídos.

Esta parte consiste, então, na codificação de n vetores de chaves aleatórias, o chamado cromossoma. Após descodificação, deve ser definido um critério de paragem. São então geradas soluções e calculado o valor do *Fitness* (ver Secção 3.2.1) que permite organizar e classificar as soluções em ELITE ou NÃO ELITE, mediante o valor da Função Objetivo. As soluções consideradas como ELITE (melhores soluções) são copiadas para a próxima população. Em seguida, é introduzida uma população aleatória externa denominada mutante. É, então, feito o *crossover* entre a população ELITE e NÃO ELITE, de forma a dar origem a descendentes, que serão copiados para a geração seguinte. Ocorre, assim, uma evolução da população.

A parte dependente do problema é implementada de acordo com o caso em estudo. Consiste na tradução dos valores da chave aleatória gerada anteriormente numa solução para o problema e na avaliação do *fitness* dessa solução. Esta tradução é feita utilizando um chamado descodificador, que tem como função a descodificação de chaves aleatórias em soluções.

3.2.1 *Fitness* e Admissibilidade da solução

O *fitness* de uma solução é o valor do lucro total obtido para o problema, ou seja, o valor da Função Objetivo do modelo previamente apresentado. Para distinguir as diferentes reservas, foram considerados três diferentes perfis de aluguer: a reserva dita normal, o *upgrade* e o *downgrade*. O lucro real de cada um destes perfis é a diferença entre a receita obtida em cada reserva alocada e o custo resultante da transferência em vazio do veículo, caso necessário, para a estação pretendida pelo cliente. De forma a garantir as bases para a comparação com os resultados da literatura, o cálculo de lucro baseou-se na função objetivo do modelo da Secção 3.1.1. Para a sua construção considerou-se o seguinte para cada perfil:

- Reserva normal – o grupo requerido pelo cliente é o mesmo que o atribuído à reserva; o valor resultante deste tipo de reservas é o lucro da reserva menos o custo de transferência em vazio (se necessário) entre a estação onde o veículo se encontrava antes da reserva e a estação de início da reserva. Corresponde à primeira parcela da função;
- *Upgrade* – o grupo de veículo pretendido pelo cliente é de grupo inferior ao atribuído à reserva; o valor resultante deste tipo de reservas é o lucro correspondente menos o custo de transferência da estação onde o veículo está disponível para a estação onde se iniciará a reserva, sendo ainda decrescido de uma unidade de forma a garantir que esta opção é preterida face à reserva normal. Corresponde à segunda parcela da função;

- *Downgrade* – o grupo de veículo pretendido pelo cliente é superior ao grupo do atribuído à reserva; de forma a garantir que esta opção é apenas utilizada como último recurso, estas reservas apenas apresentam lucro de uma unidade monetária. Corresponde à terceira parcela.

Algoritmo 1: CÁLCULO DA FUNÇÃO *Fitness*

Entrada: lista de reservas alocadas

Saída: Valor do lucro obtido (fitness)

```

1 fitness ← 0
2 início
3   para cada veículo  $v$  em  $\mathcal{V}$  faça
4     tipo de reserva ← determinar tipo de reserva( $res_0, v$ )
5     se tipo de reserva = normal então
6       fitness ← fitness + lucrores0 - custoestação  $v$ , início  $res_0$ 
7     fim
8     senão se tipo de reserva = upgrade então
9       fitness ← fitness + lucrores0 - custoestação  $v$ , início  $res_0$  - 1
10    fim
11    senão
12      fitness ← fitness + 1
13    fim
14  fim
15  para cada reserva  $i$  no veículo  $v$  faça
16    tipo de reserva ← determinar tipo de reserva( $res_i, v$ )
17    se tipo de reserva = normal então
18      fitness ← fitness + lucroresi - custofim  $res_{i-1}$ , início  $res_i$ 
19    fim
20    senão se tipo de reserva = upgrade então
21      fitness ← fitness + lucroresi - custofim  $res_{i-1}$ , início  $res_i$  - 1
22    fim
23    senão
24      fitness ← fitness + 1
25    fim
26  fim
27 fim
```

Apresenta-se no Algoritmo 1 o cálculo do valor do *Fitness*, baseado no tipo de reserva. A função *determinar tipo de reserva* determina se uma reserva é normal, *upgrade* ou *downgrade*, isto é, se é alocada ao veículo pretendido, a um de uma gama superior ou a um de uma gama inferior. Consideram-se duas situações distintas: a alocação da primeira reserva ao veículo e a alocação das restantes. O custo apresentado é relativo à transferência

em vazio dos veículos e é calculado de acordo com a estação de localização do veículo para a estação de início da reserva. Na alocação da primeira reserva ao veículo, é considerada a localização inicial do mesmo. Quando existe mais do que uma reserva já alocada ao veículo, a estação inicial passa então a ser a estação final da última reserva alocada.

Admissibilidade das soluções

O modelo proposto por Oliveira et al. (2014) é um modelo de fluxo em rede e, como tal, inclui restrições que não estão relacionadas com o problema em si, mas sim com a estrutura do modelo e com características das variáveis de decisão. Para o processo de alocação de reservas a veículos em si, o BRKGA não necessita da implementação destas restrições.

Assim, para ser testada a possibilidade de alocação de uma reserva a um veículo específico, deve ser verificado que:

- O veículo está disponível no instante de início da reserva, como indica a Restrição 3.5 no modelo, considerando o tempo que poderá ser necessário para o trazer para a estação de início (tempo de transferência em vazio);
- O período de duração da reserva não coincide com um período de indisponibilidade temporária do veículo, usado para a manutenção da frota *impros* (Restrição 3.10);
- O período de duração da reserva não coincide com os períodos de indisponibilidade permanente – *block date* (Restrição 3.9);

É ainda garantida a prioridade das reservas confirmadas face às não confirmadas (Restrição 3.11).

3.2.2 Descodificação dos Cromossomas

Decodificador de Alocação

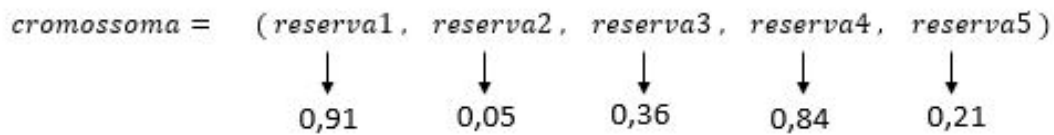
A primeira tradução das chaves aleatórias em solução foi chamada de *Decodificador de Alocação*. Este primeiro método consiste na divisão do intervalo $[0...1]$ em $V + 1$ intervalos, sendo V o número de veículos existentes na frota. O intervalo extra é considerado para representar a não-alocação de uma reserva, quando não há nenhum veículo disponível.

Exemplo de decodificador de Alocação Considerem-se 5 reservas (reserva 1, ..., reserva 5) e 3 veículos (A, B, C). O processo de decodificação inicia-se com o cromossoma que assume a seguinte forma:

$$cromossoma = (reserva1, reserva2, reserva3, reserva4, reserva5)$$

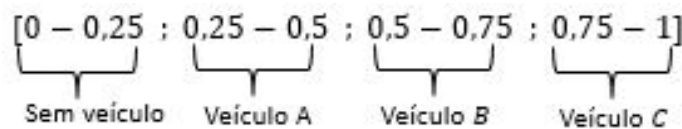
É atribuído um número aleatório pertencente ao intervalo $[0...1]$ a cada reserva pertencente ao cromossoma, segundo a Figura 3.2.

Figura 3.2: Codificação – Exemplo atribuição chave aleatória



Seguidamente, é feita a divisão do intervalo $[0...1]$ em 4 partes iguais (como mostra a Figura 3.3), onde cada intervalo corresponde a um veículo, exceto o primeiro, que representa a não-alocação da reserva.

Figura 3.3: Compartimentação do intervalo



A atribuição das reservas (genes do cromossoma) aos veículos (intervalos) é feita conforme o intervalo em que está incluído o valor da chave aleatória (Figura 3.4)

Figura 3.4: Atribuição veículo-reserva



Admissibilidade das soluções Para garantir a admissibilidade da solução, as restrições mencionadas na Secção 27 devem ser verificadas quando se faz cada uma das alocações reserva-veículo. Para minimizar o tempo de processamento, as verificações foram implementadas de forma a que ocorra uma falha rápida. Entende-se por falha rápida a interrupção da verificação das restrições, quando uma delas não é verificada, pois a solução torna-se imediatamente inadmissível.

Testes preliminares Para os testes preliminares foram utilizadas instâncias presentes na literatura, descritas posteriormente na Secção 4.2, tendo sido os testes corridos no computador com características descritas na Secção 4.1. Testes preliminares da implementação do Decodificador de Alocação para a totalidade das reservas presentes em cada instância permitiram verificar que o método conseguia um número muito reduzido (e por vezes nulo) de soluções admissíveis. Por exemplo, para a Instância 1, após 26 horas, ainda não tinha

sido encontrada uma solução admissível. Com este tipo de descodificação e com a dimensão das instâncias utilizadas, o BRKGA não se mostrou capaz de responder eficazmente ao problema.

Face aos resultados preliminares desfavoráveis na resolução das instâncias por inteiro, foi decidido compartimentar as instâncias e resolvê-las num horizonte temporal rolante, com *overlap* entre partições, para melhorar a alocação de reservas (ver Secção 3.3). Também esta solução não atingiu os objetivos pretendidos, uma vez que apenas permitia o processamento de 8 reservas (em 50) em cada janela temporal, o que, comparando com os resultados disponibilizados na literatura, é um mau resultado. Sendo as instâncias constituídas por mais de mil reservas, tal solução não vai de encontro aos objetivos do estudo.

Descodificador de Ordenação + Heurística de Alocação

Face aos resultados insatisfatórios obtidos pelo Descodificador de Alocação, uma nova abordagem foi implementada. Implementou-se então em duas partes: ordenação das reservas para posterior alocação com chamado Descodificador de Alocação e uma segunda parte de alocação das reservas pela ordem previamente estabelecida – Heurística de Alocação.

Neste segundo método, a estrutura do cromossoma permanece inalterada, sendo igualmente composto pelo número de reservas. Como processo de codificação, é atribuído a cada reserva um número aleatório pertencente ao intervalo $[0...1]$. Mediante esses valores, é feita uma ordenação por ordem crescente dessas chaves aleatórias, e guardada a posição inicial, antes de proceder à ordenação numérica. Ficam então as reservas temporariamente desordenadas, mas será a ordem de ordenação numérica que determinará a ordem de posterior tentativa de alocação ao veículo.

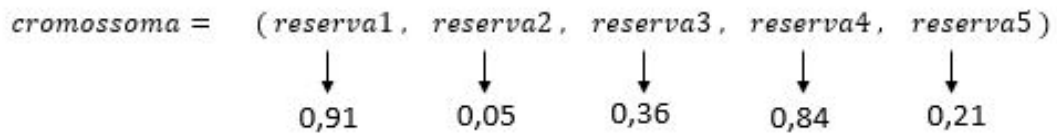
Segue-se uma heurística construtiva de alocação: com as reserva já ordenadas pelas chaves geradas pelo BRKGA, procede-se à sua alocação nos veículos, pela ordem gerada. Para cada reserva, são percorridos os veículos até encontrar um que cumpra todas as condições, tornando-se numa solução admissível. Quando há possibilidade de alocação, esta é efetuada de imediato. Esta heurística é uma forma rápida de conseguir soluções sempre admissíveis, sendo a aleatoriedade introduzida pela ordem pela qual as reservas são incluídas.

Exemplo de Descodificador de Ordenação + Heurística de Alocação Considerando 5 reservas (reserva1, ... , reserva 5) e 3 veículos (A, B, C), o cromossoma toma a seguinte forma:

$cromossoma = (reserva1, reserva2, reserva3, reserva4, reserva5)$

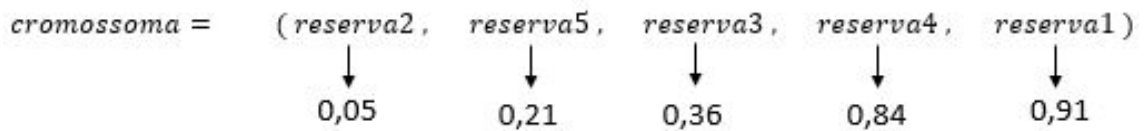
É então atribuído a cada reserva um número aleatório entre $[0...1]$ (Figura 3.5).

Figura 3.5: Codificação – Exemplo atribuição chave aleatória



Como primeiro passo, é feita uma ordenação numérica dos genes (reservas) do cromossoma (Figura 3.6)

Figura 3.6: Codificação – Exemplo de ordenação



Procede-se, então, à alocação das reservas na seguinte ordem: 2,5,3,4,1, conforme a satisfação das restrições cumpridas. À primeira verificação a reserva é atribuída de imediato ao veículo.

Admissibilidade de Solução Neste tipo de decodificação, e ao contrário da anterior, não há soluções não admissíveis. Se não forem verificadas todas as restrições em nenhum veículo, então não será atribuído nenhum veículo à reserva em questão.

3.3 Resolução em horizonte rolante

Sendo cada gene do cromossoma correspondente a uma reserva e tendo cada instância milhares de reservas, o cromossoma assume grandes dimensões, como no primeiro decodificador apresentado, o que exige elevados tempos de processamento. Efetuou-se então a divisão das reservas, para cada instância, em partes iguais, permitindo desta forma obter cromossomas de tamanho muito inferior, diminuindo os tempos de processamento. Para tal ser possível é necessário, a implementação em horizonte rolante. Com vista em fazer comparação dos resultados com Oliveira et al. (2014), compartimentaram-se as reservas em 50 unidades com sobreposição de 15. As reservas de sobreposição servem para a ligação entre as partições em blocos, ou seja, para as reservas da partição seguintes serem afetadas pelas reservas anteriores.

3.3.1 Resultados preliminares – Instância Completa

Para justificar a necessidade de utilização do horizonte rolante realizaram-se testes preliminares do Decodificador de Alocação + Heurística de Ordenação sem partição das

reservas, ou seja, a janela em teste é a totalidade das reservas da instância. A instância utilizada para teste foi a Instância 1 (ver características da instância na Secção 4.2), com o computador utilizado para a análise de sensibilidade cujas características são descritas na secção 4.1.

Tabela 3.1: Resultados preliminares para resolução de uma instância completa

Instância	Nr Reservas Alocadas	Tempo de Processamento (min)	Função Objetivo
1	117	166	140.704

Os resultados obtidos não foram de encontro aos esperados, especialmente no que se refere ao tempo de resolução, confirmando-se a necessidade da utilização do horizonte rolante. Ao mesmo tempo, o método não foi capaz de alocar um número satisfatório de reservas, o que se traduziu em valores baixos alcançados pela Função Objetivo. Sendo o tempo de processamento um dos principais parâmetros em estudo, também este se revelou mais elevado do que o esperado. É, então, possível concluir que esta não é a melhor solução para a resolução do problema, uma vez que demora demasiado tempo a alcançar resultados medíocres, sendo assim justificada a utilização do horizonte rolante de resolução.

Capítulo 4

Testes e Resultados

Neste capítulo serão apresentados os motivos pelos quais foi feita a análise de sensibilidade e os *insights* daí obtidos, e, ainda, descritas as condições dos testes. De seguida, serão apresentados os resultados computacionais obtidos e comparados com resultados presentes na literatura.

4.1 Análise de sensibilidade

Na implementação do método considerou-se pertinente fazer uma análise de sensibilidade, de forma a testar o impacto que a variação de determinados parâmetros poderiam ter na Função Objetivo. Dos parâmetros listados por (Gonçalves e Resende, 2014), foi decidido estudar o impacto da variação do valor de p – população; p_e – tamanho da partição ELITE; p_m – tamanho da partição MUTANTE.

Utilizando o horizonte rolante descrito na Secção 3.3 que considera o processamento de 50 reservas de cada vez e considerando as recomendações de Gonçalves e Resende (2014), escolheram-se as seguintes variações:

- População (p) – o tamanho da população foi variado entre $p = 500$, $p = 1000$ e $p = 2500$ para verificar o impacto do aumento da população na Função Objetivo;
- Tamanho da partição ELITE (p_e) – sendo recomendado pelos autores variar este parâmetro entre 0.1 e 0.25, optou-se pelo seu valor mínimo (0.1), valor médio (0.175) e valor máximo (0.25);
- Tamanho da partição MUTANTE (p_m) – a recomendação de variação para esta partição é entre 0.05 e 0.2, sendo então novamente escolhido o valor mínimo (0.05), o valor médio (0.12) e o valor máximo (0.2).

De entre os parâmetros recomendados pelos autores, estes foram escolhidos para a análise de sensibilidade por estarem mais relacionados com a parte independente do problema, isto é, com a estrutura dos cromossomas e a estrutura da população. Os restantes

parâmetros recomendados para variação (ρ_a , k_r , π , k_p e k_m) estão mais relacionados com a parte dependente do problema e a evolução da meta-heurística. Nos testes efetuados, devido ao tempo de execução do programa e ao número elevado de combinações resultantes da variação dos parâmetros referidos, apenas foi utilizada a Instância 1 (ver Secção 4.2) para a análise de sensibilidade.

A máquina utilizada para esta análise foi um computador da marca ASUS, modelo S46CM, com um processador Intel(R) Core(TM) i7-3517U CPU @ 1.90GHz 2.4GHz, memória instalada (RAM) 6Gb, com o sistema operativo *Windows 10 Home* de 64bits. O programa foi escrito em linguagem C++, corrido em *Visual Studio 2013*.

Resultados obtidos

Tendo como principal objetivo a maximização do lucro total obtido, iniciar-se-á a análise dos diferentes valores da Função Objetivo para as diferentes populações. Para $p = 500$ (Tabela 4.1) e para diferentes valores da população de (p_e) e (p_m), o valor da Função Objetivo permanece praticamente inalterado. O mesmo acontece para $p = 1000$ (Tabela 4.2) e $p = 2500$ (Tabela 4.3). Desde já, o método demonstra robustez em termos dos resultados obtidos para as variações recomendadas dos parâmetros.

Tabela 4.1: Função Objetivo para $p = 500$

$p_e \backslash p_m$	0.05	0.12	0.2
0.1	295.456	291.992	294.571
0.175	294.980	290.536	292.021
0.25	293.429	294.589	290.933

Tabela 4.2: Função Objetivo para $p = 1000$

$p_e \backslash p_m$	0.05	0.12	0.2
0.1	291.000	292.235	289.701
0.175	290.692	288.983	292.754
0.25	288.367	292.587	289.132

Tabela 4.3: Função Objetivo para $p = 2500$

$p_e \backslash p_m$	0.05	0.12	0.2
0.1	291.975	292.052	293.510
0.175	291.230	291.941	288.898
0.25	291.157	292.483	290.776

Para cada uma das populações foi ainda relevante testar o tempo de processamento pois, além de maximizar o valor da Função Objetivo, um outro objetivo deste trabalho

é minimizar o tempo que o programa demora a construir o planeamento da atribuição veículo-reserva.

Para as diferentes populações consideradas, verificou-se um aumento do tempo à medida que se aumentava a população. Os menores valores verificam-se para $p = 500$ (Tabela 4.4), encontrando-se aproximadamente nos 30 minutos. Aumentando a população para $p = 1000$ (Tabela 4.5), verifica-se que o tempo de processamento já sofre um aumento para aproximadamente o dobro, aumentando ainda mais consideravelmente para $p = 2500$ (Tabela 4.6).

Tabela 4.4: Tempo de processamento (min) para $p = 500$

$p_e \backslash p_m$	0.05	0.12	0.2
0.1	32.1	32.37	32.2
0.175	32.03	34.8	29.83
0.25	26.9	27.5	2060

Tabela 4.5: Tempo de processamento (min) para $p = 1000$

$p_e \backslash p_m$	0.05	0.12	0.2
0.1	66.2	40.25	67.82
0.175	59.42	34.8	60.45
0.25	55.3	54.8	55.15

Tabela 4.6: Tempo de processamento (min) para $p = 2500$

$p_e \backslash p_m$	0.05	0.12	0.2
0.1	162	131.9	163
0.175	120.63	120.1	120.9
0.25	109.92	111.8	110.48

Apesar de a Função Objetivo ser diretamente proporcional ao número de reservas alocadas, estas foram ,ainda, alvo de análise para as diferentes populações: $p = 500$ (Tabela 4.7), $p = 1000$ (Tabela 4.8) e $p = 2500$, (Tabela 4.9). Como de esperar, também o número de reservas alocadas se mantém sem variações significativas à medida que se aumenta o tamanho da população.

É ainda possível consultar os resultados dos tempos de transferência em vazio para estas soluções no Anexo A. Estes valores mostraram uma evolução semelhante e de encontro ao esperado, após análise dos indicadores acima mencionados.

Tabela 4.7: Número de reservas alocadas para $p = 500$

$p_e \backslash p_m$	0.05	0.12	0.2
0.1	219	215	215
0.175	221	214	218
0.25	219	223	215

Análise e conclusões

Com esta análise de sensibilidade, foi possível concluir que o BRKGA é um método bastante robusto, uma vez que os indicadores testados (Função Objetivo, reservas alocadas e tempo de transferência em vazio) para variações do tamanho da população, da partição da população ELITE e partição da população MUTANTE não apresentaram variações significativas. O aumento da tamanho da população não induz grande variação nestes parâmetros, exceto no tempo de processamento que, à medida que aumenta a população, aumenta também. É, então, possível concluir que o melhor tamanho de população a utilizar para os testes computacionais deverá ser $p = 500$. Para valores semelhantes de Função Objetivo, tempo de transferência em vazio e reservas alocadas será necessário um menor tempo de processamento.

Dado que se concluiu que os outros parâmetros não causam variações significativas nos indicadores mais relevantes, foram escolhidos os valores médios para correr os testes computacionais mais intensivos.

Foram observadas ao longo dos repetidos testes da Análise de Sensibilidade algumas oscilações inesperadas relativamente aos tempos de processamento. Estes tempos podem atingir valores mais elevados do que os valores esperados devido às limitações do computador em que foram testados pois, apesar dos testes serem feitos com o mínimo de tarefas em execução no computador, é necessária uma partilha de recursos e quanto maior for o número de tarefas em execução, maior será o tempo decorrido.

4.2 Resultados Computacionais

Devido ao condicionado desempenho verificado no computador inicialmente utilizado, mudou-se a máquina de teste para um servidor alojado no INESC-TEC com as seguintes

Tabela 4.8: Número de reservas alocadas para $p=1000$

$p_e \backslash p_m$	0.05	0.12	0.2
0.1	211	216	214
0.175	214	215	214
0.25	222	217	214

Tabela 4.9: Número de reservas alocadas para $p = 2500$

$p_e \backslash p_m$	0.05	0.12	0.2
0.1	220	219	221
0.175	218	221	213
0.25	218	216	211

características: 2 processadores Intel(R) Xeon(R) CPU X5690 @ 3.46 GHz, com 48 Gb de RAM, e sistema operativo Windows 7 Professional de 64-bit.

Com uma máquina mais rápida e capaz, foi então possível testar as 20 instâncias disponibilizadas em [Oliveira et al. \(2014\)](#) de forma a ser possível comparar com os resultados obtidos por estes autores. São apresentadas na Tabela 4.10 algumas das características mais relevantes das instâncias utilizadas nos testes.

Variação dos parâmetros

Com novos recursos, aumentou-se a rapidez de processamento, e foram, assim, consideradas novas variações de diferentes parâmetros. Considerando a influência de parâmetros anteriores já testados, foram, então, fixados os valores para os quais se obtiveram anteriormente os melhores resultados, mantendo estes constantes e introduzindo novas variações de outros parâmetros. Foram considerados relevantes o número máximo de gerações (MaxGens) e a *Seed*. Para o valor da população $p = 500$, proporção da população ELITE $p_e = 0.175$ e proporção da população MUTANTE $p_m = 0.12$ (valores que permitiram anteriormente alcançar os melhores resultados), efetuaram-se os mesmos testes para valores máximos de gerações MaxGens=50 e MaxGens=1000. Sendo o BRKGA um algoritmo não determinístico, consideraram-se, ainda, para cada configuração, 20 valores aleatórios diferentes para a *seed*.

A análise incidiu nos valores da Função Objetivo e no tempo de processamento necessário para alcançar esses resultados, com número máximo de gerações de 50 (Tabela 4.11) e 1000 (Tabela 4.12). Nestas tabelas podemos observar (após as vinte corridas com *seeds* diferentes para cada instância) o valor médio da função objetivo, o seu desvio padrão, o tempo médio de corrida, o valor máximo da função objetivo e o tempo em que esse valor máximo foi obtido.

Compararam-se os resultados obtidos para o número máximo de gerações de 50 e 1000 (Tabela 4.13). Esta tabela apresenta a taxa de variação percentual quando se aumenta o número máximo de gerações de 50 para 1000 dos seguintes indicadores: Função Objetivo média, tempo de processamento médio, Função Objetivo máxima, tempo para obter o valor da Função Objetivo máxima, Função Objetivo mínima e tempo para obter o valor da Função Objetivo mínimo. Para ambos os valores do número máximo de gerações testados foram utilizados os mesmos valores de *Seed*. É então possível verificar que, como esperado, o tempo de processamento necessário para correr 50 gerações é muito inferior

Tabela 4.10: Características das instâncias

Instância	Número de Reservas	Número de veículos	Número de <i>impros</i>
1	1149	39	5
2	2683	20	1
3	2664	28	3
4	2379	26	0
5	1265	36	2
6	1707	27	0
7	1238	34	0
8	1555	26	1
9	1491	37	0
10	2440	25	4
11	2136	27	0
12	1287	31	4
13	2521	26	3
14	2363	35	2
15	2684	26	0
16	2696	29	2
17	1758	39	1
18	2012	27	1
19	2228	27	0
20	1976	39	0

Tabela 4.11: Resultados para número máximo de gerações MaxGens=50

Instância	FO (\bar{x})	FO (σ)	Tempo[s] (\bar{x})	FO (max)	Tempo[s] (FO max)
1	290.535	2123	13	294.637	13
2	258.739	937	23	260.709	24
3	292.758	676	24	294.023	24
4	225.671	827	21	226.791	20
5	318.958	1488	13	321.412	13
6	219.508	953	15	221.345	14
7	230.500	1081	13	232.810	12
8	249.686	969	13	251.737	13
9	263.232	1573	15	266.339	15
10	224.872	1381	21	227.589	21
11	258.070	2372	18	261.621	20
12	248.660	1313	12	251.757	13
13	236.774	2015	22	239.662	22
14	303.415	2152	24	306.893	24
15	249.202	2233	22	253.197	24
16	307.493	1366	25	310.086	25
17	332.107	866	19	333.806	18
18	255.190	1231	17	256.920	17
19	229.143	1326	19	231.775	19
20	331.091	1705	21	334.034	21

Tabela 4.12: Resultados para o número máximo de gerações MaxGens=1000

Instância	FO (\bar{x})	FO (σ)	Tempo[s] (\bar{x})	FO (max)	Tempo[s] (FO max)
1	290.978	2114	243	296.058	241
2	259.277	915	437	261.080	437
3	293.649	710	441	295.096	442
4	226.576	663	402	227.808	394
5	319.297	1430	247	321.357	246
6	220.700	750	276	221.679	279
7	230.329	1383	239	233.414	240
8	249.562	1044	233	252.908	228
9	263.838	1326	286	266.645	283
10	225.317	1462	388	227.822	386
11	258.274	2417	344	261.916	339
12	248.691	1257	230	251.487	227
13	237.465	1983	404	240.699	404
14	303.833	2116	448	307.717	444
15	249.623	2086	417	252.920	419
16	307.688	1279	455	309.364	455
17	332.673	644	349	333.793	343
18	255.512	1265	319	257.460	309
19	229.441	1066	359	232.527	384
20	331.807	1807	387	335.017	381

ao tempo necessário para correr 1000 (cerca de 95% inferior), tempos estes que não influenciam significativamente o valor da Função Objetivo, cujos resultados permanecem aproximadamente iguais para os dois valores de gerações máximas.

Tabela 4.13: Taxa de variação de MaxGens=50 para MaxGens=1000

Instância	FO (\bar{x})	Tempo(\bar{x})	FO (max)	Tempo[s] FO (max)	FO(min)	Tempo[s] FO (min)
1	0%	-95%	0%	-95%	0%	-94%
2	0%	-95%	0%	-95%	0%	-94%
3	0%	-95%	0%	-95%	0%	-95%
4	0%	-95%	0%	-95%	0%	-95%
5	0%	-95%	0%	-95%	0%	-94%
6	-1%	-95%	0%	-95%	0%	-94%
7	0%	-95%	0%	-95%	0%	-95%
8	0%	-95%	0%	-94%	0%	-95%
9	0%	-95%	0%	-95%	-1%	-95%
10	0%	-95%	0%	-95%	0%	-95%
11	0%	-95%	0%	-94%	0%	-95%
12	0%	-95%	0%	-94%	0%	-95%
13	0%	-95%	0%	-95%	-1%	-95%
14	0%	-95%	0%	-95%	0%	-95%
15	0%	-95%	0%	-94%	0%	-95%
16	0%	-95%	0%	-95%	0%	-94%
17	0%	-95%	0%	-95%	0%	-95%
18	0%	-95%	0%	-94%	0%	-95%
19	0%	-95%	0%	-95%	0%	-95%
20	0%	-95%	0%	-94%	0%	-95%

Para avaliar quantas gerações foram necessárias para alcançada a melhor solução, novamente para os dois valores máximos de gerações, foi analisada a geração média para a qual cada instância encontra a melhor solução (Tabela 4.14). Tanto para o número máximo de gerações de 50 como de 1000, o melhor resultado é alcançado muito cedo. Para MaxGens=50, são necessárias apenas 15 gerações, e para MaxGens=1000 são necessárias apenas 60. Apesar de estes valores mostrarem que o método converge cedo para a melhor solução que encontra, esta pode não ser a solução ideal, pois o método não está a introduzir a aleatoriedade esperada. O motivo pelo qual isto acontece poderá ser o facto da Heurística Construtiva estar a limitar a aleatoriedade esperada do *framework* BRKGA.

Para a população $p = 500$, e variando o valor da *Seed*, utilizaram-se os valores médios obtidos para comparação com os resultados de Oliveira et al. (2014). O primeiro parâmetro a ser comparado com os autores é o valor da Função Objetivo (Tabela 4.15). Os resultados obtidos pelo BRKGA foram cerca de 7% inferiores. Uma vez que o valor da Função Objetivo é diretamente proporcional ao número de reservas alocadas, este valor foi também ligeiramente inferior (cerca de 16%) (consultar Anexo A). Pode ainda ser consultada neste anexo a comparação do tempo total de transferências em vazio.

Tabela 4.14: Geração média na qual foi alcançada a melhor solução

Instância	MaxGens = 50	MaxGens= 1000
1	12	63
2	4	15
3	6	17
4	7	26
5	15	95
6	9	40
7	10	40
8	10	36
9	9	50
10	4	12
11	6	16
12	11	60
13	5	24
14	7	28
15	4	15
16	5	17
17	10	44
18	7	30
19	6	21
20	9	41

Segundo a literatura, uma das principais vantagens da utilização do BRKGA é a sua rapidez para encontrar uma solução. Na implementação testada foi confirmado este comportamento (Tabela 4.16) dado que, comparando com os resultados dos autores, foi conseguida uma redução do tempo de processamento em cerca de 96%. É aqui onde se verifica a mais considerável melhoria.

Tabela 4.15: Comparação do valor da Função Objetivo com [Oliveira et al. \(2014\)](#)

Instância	Função Objetivo		
	Oliveira et al. (2014)	BRKGA (max) MaxGens=50	BRKGA vs. Oliveira et al. (2014)
1	291.219	294.637	1%
2	272.822	260.709	-4%
3	307.344	294.023	-4%
4	255.208	226.791	-11%
5	321.569	321.412	0%
6	237.004	221.345	-7%
7	247.569	232.810	-6%
8	271.140	251.737	-7%
9	263.548	266.339	1%
10	240.097	227.589	-5%
11	278.634	261.621	-6%
12	260.340	251.757	-3%
13	254.532	239.662	-6%
14	329.680	306.893	-7%
15	262.147	253.197	-3%
16	319.811	310.086	-3%
17	339.137	333.806	-2%
18	266.758	256.920	-4%
19	244.401	231.775	-5%
20	345.103	334.034	-3%

Tabela 4.16: Comparação do tempo de processamento com [Oliveira et al. \(2014\)](#)

Instância	Tempo processamento[s] - FO(max)		
	Oliveira et al. (2014)	BRKGA	BRKGA vs. Oliveira et al. (2014)
1	314	13	-96%
2	521	24	-95%
3	597	24	-96%
4	540	20	-96%
5	718	13	-98%
6	418	14	-97%
7	444	12	-97%
8	365	13	-96%
9	474	15	-97%
10	638	21	-97%
11	581	20	-97%
12	464	13	-97%
13	704	22	-97%
14	721	24	-97%
15	731	24	-97%
16	861	25	-97%
17	468	18	-96%
18	449	17	-96%
19	484	19	-96%
20	606	21	-97%

Capítulo 5

Conclusão

Na resolução do problema de alocação veículo-reserva para uma empresa de *rent-a-car*, o BRKGA provou ser um método robusto, conseguindo resultados ao nível dos da literatura, e particularmente eficaz no tempo de processamento. De facto, ao longo dos testes efetuados, esta meta-heurística provou ser capaz de alcançar resultados muito próximos dos resultados presentes na literatura, diminuindo o tempo de processamento em cerca de 25 vezes. Estando já publicadas boas soluções para este problema, foram utilizadas as mesmas instâncias para comparação dos resultados. Diferentes adaptações do *framework* ao problema em questão foram testadas. O Descodificador de Ordenação aliado a uma Heurística de Alocação, testado em horizonte rolante, foi capaz de alcançar as melhores soluções para o problema.

Mantendo sempre o foco na maximização do lucro obtido, diretamente relacionado com o aumento do número de reservas alocadas, é possível concluir que o BRKGA, apesar de não ter melhorado as soluções existentes, alcançou valores muito próximos mais rapidamente. Neste tipo de empresas, um rápido tempo de resposta na alocação das reservas aos veículos é fundamental, pois com um processo de alocação rápido têm sempre disponível o calendário de marcações atualizado, estando disponíveis para imediata alocação de novas reservas, não comprometendo a qualidade do serviço.

O facto de não terem sido alcançados melhores valores de lucro poder-se-á dever ao facto da Heurística Construtiva (determinística) limitar a vantagem do BRKGA, ao reduzir a aleatoriedade introduzida. Assim, como trabalho futuro, seria relevante considerar a implementação de diferentes heurísticas construtivas, nomeadamente heurísticas que introduzissem aleatoriedade ao problema.

Anexo A

Indicadores secundários de análise

Tabela A.1: Tempo de transferência em vazio para $p = 500$

$p_e \backslash p_m$	0.05	0.12	0.2
0.1	2084	2296	2035
0.175	2134	2080	2103
0.25	2203	2358	2060

Tabela A.2: Tempo de transferência em vazio para $p = 1000$

$p_e \backslash p_m$	0.05	0.12	0.2
0.1	2241	2292	2135
0.175	2172	1921	2290
0.25	2242	2259	2149

Tabela A.3: Tempo de transferência em vazio para $p = 2500$

$p_e \backslash p_m$	0.05	0.12	0.2
0.1	2241	2292	2135
0.175	2172	1921	2290
0.25	2242	2259	2149

Tabela A.4: Comparação do número de reservas alocadas com [Oliveira et al. \(2014\)](#)

Instância	Reservas Alocadas		
	Oliveira et al. (2014)	BRKGA	BRKGA vs. Oliveira et al. (2014)
1	222	215	-3%
2	218	194	-11%
3	221	186	-16%
4	203	170	-16%
5	262	241	-8%
6	198	170	-14%
7	199	176	-12%
8	196	159	-19%
9	207	209	1%
10	203	168	-17%
11	207	168	-19%
12	202	189	-7%
13	217	187	-14%
14	256	218	-15%
15	185	162	-13%
16	231	210	-9%
17	254	240	-5%
18	200	177	-11%
19	205	175	-15%
20	280	246	-12%

Tabela A.5: Comparação do tempo de transferência em vazio com [Oliveira et al. \(2014\)](#)

Instância	Transferência em Vazio [horas]		
	Oliveira et al. (2014)	BRKGA	BRKGA vs. Oliveira et al. (2014)
1	1984	2103	6%
2	1715	1920	12%
3	1664	1863	12%
4	1824	1823	0%
5	1710	1907	12%
6	1723	1623	-6%
7	1997	1634	-18%
8	1633	1407	-14%
9	1966	1649	-16%
10	1976	2045	3%
11	1777	1475	-17%
12	1526	1683	10%
13	1606	1538	-4%
14	2236	2054	-8%
15	1688	1680	0%
16	2249	1916	-15%
17	2066	2048	-1%
18	1786	1690	-5%
19	1796	1867	4%
20	2091	2197	5%

Bibliografia

Beaujon, G. J. e M. A. Turnquist

1991. A Model for Fleet Sizing and Vehicle Allocation. *Transportation Science*, 25(1):19–45.

Bojović, N. J.

2002. A general system theory approach to rail freight car fleet sizing. *European Journal of Operational Research*, 136(1):136–172.

Carroll, W. J. e R. C. Grimes

1995. Evolutionary Change in Product Management: Experiences in the Car Rental Industry. *Interfaces*, 25(5):84–104.

Crainic, T. G.

2000. Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288.

Crainic, T. G., J.-a. Ferland, e J.-M. Rousseau

1984. A Tactical Planning Model for Rail Freight Transportation. *Transportation Science*, 18(2):165–184.

Ernst, A. T., E. O. Gavriliouk, e L. Marquez

2011. An efficient Lagrangean heuristic for rental vehicle scheduling. *Computers and Operations Research*, 38(1):216–226.

Fink, A. e T. Reiners

2006. Modeling and solving the short-term car rental logistics problem. *Transportation Research Part E: Logistics and Transportation Review*, 42(4):272–292.

Gonçalves, J. F. e M. G. Resende

2004. An evolutionary algorithm for manufacturing cell formation. *Computers & Industrial Engineering*, 47(2):247–273.

Gonçalves, J. F. e M. G. Resende

2014. Random-key genetic algorithms. *Handbook of Heuristics*.

Gonçalves, J. F. e M. G. C. Resende

2010. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.

Grasas, A., H. Ramalhinho, L. S. Pessoa, M. G. C. Resende, I. Caballé, e N. Barba

2014. On the improvement of blood sample collection at clinical laboratories. 2003:9.

Lalla-Ruiz, E., J. L. González-Velarde, B. Melián-Batista, e J. M. Moreno-Vega

2014. Biased random key genetic algorithm for the Tactical Berth Allocation Problem. *Applied Soft Computing Journal*, 22:60–76.

Li, Z. e F. Tao

2010. On determining optimal fleet size and vehicle transfer policy for a car rental company. *Computers & Operations Research*, 37(2):341–350.

Marques, I., M. E. Captivo, e M. Vaz Pato

2014. Scheduling elective surgeries in a Portuguese hospital using a genetic heuristic. *Operations Research for Health Care*, 3(2):59–72.

Milenković, M. S., N. J. Bojović, L. Švadlenka, e V. Melichar

2015. A stochastic model predictive control to heterogeneous rail freight car fleet sizing problem. *Transportation Research Part E: Logistics and Transportation Review*, 82:162–198.

Oliveira, B. B.

2013. Car rental logistics : a metaheuristic and a matheuristic for the vehicle-reservation assignment problem. MSc Thesis. *Universidade do Porto, Faculdade de Engenharia*.

Oliveira, B. B., M. A. Carravilla, J. F. Oliveira, e F. M. Toledo

2014. A relax-and-fix-based algorithm for the vehicle-reservation assignment problem in a car rental company. *European Journal of Operational Research*, 237(2):729–737.

Pachon, J., E. Iakovou, e I. Chi

2006. Vehicle fleet planning in the car rental industry. *Journal of Revenue and Pricing Management*, 5(3):221–236.

Pachon, J. E., E. Iakovou, C. Ip, e R. Aboudi

2003. A Synthesis of Tactical Fleet Planning Models for the Car Rental Industry. *IIIE Transactions*, 35(9):907–916.

Silva, R. M. A., M. G. C. Resende, e P. Pardalos

2015. A Python/C++ Library for bound-constrained global optimization using Biased Random-Key Genetic Algorithm. *Journal of Combinatorial Optimization*, 30.3:710–728.

Spieckermann, S. e S. Voß

1995. A case study in empty railcar distribution. *European Journal of Operational Research*, 87(3):586–598.

Yaghini, M. e Z. Khandaghabadi

2013. A hybrid metaheuristic algorithm for dynamic rail car fleet sizing problem. *Applied Mathematical Modelling*, 37(6):4127–4138.